

Multiprocesorski sistemi

Pismeni ispit, 04.07.2011.

Literatura nije dozvoljena.
Ispit traje 180 minuta.

1. Navesti četiri osnovne namene paralelnih sistema. [5 poena]
2. Ukratko opisati programski model slanja poruka. [5 poena]
3. Diskutovati implikacije protokola koherencije na pisanje paralelnih programa. [10 poena]
4. Za protokol *MESI* objasniti stanja, transakcije na magistrali i akcije protokola. Nacrtati i objasniti dijagram stanja i prelaza. [15 poena]
5. Šta se postiže primenom *cache-based* protokola? Objasniti organizaciju kataloga i izračunati njegovu veličinu. Koje su prednosti i nedostaci?[10 poena]
6. Objasniti kakve su prednosti organizacije čvora kao manjeg multiprocesora u hijerarhijskim multiprocesorskim sistemima. [10 poena]
7. Koristeći POSIX niti napisati na jeziku C ili C++ program koji vrši određenu obradu nad nizom celih brojeva. Program treba da odredi i ispiše onu vrednost iz ulaznog niza koja po apsolutnoj vrednosti najmanje odstupa od aritmetičke sredine niza. Glavni program upravlja nitima i obavlja svu komunikaciju sa korisnikom. Sve niti koje glavni program stvara treba da ravnomerno učestvuju u obradi. Ako broj elemenata niza nije celobrojni umnožak broja niti, prekinuti program. Obezbediti da stvorene niti ne počnu sa radom pre nego glavna nit završi sa stvaranjem svih niti. Sinhronizaciju početka rada stvorenih niti ostvariti preko uslovnih promenljivih (`pthread_cond_t`). [15 poena]
8. Dat je multiprocesorski sistem sa 4 identična procesora, koji koristi *MSI* za održavanje koherencije keš memorije. Svaka keš memorija ima po 2 ulaza, koji su veličine jedne reči. Preslikavanje je asocijativno. Na početku su sve keš memorije prazne. Svaki upis povećava vrednost izmenjenog podatka za 1. Početne vrednosti podataka u memoriji su 0. Data je sledeća sekvenca pristupa memoriji:

1. P0,R,A0	3. P0,W,A0	5. P0,R,A0	7. P3,W,A3
2. P1,R,A0	4. P3,W,A1	6. P0,R,A1	8. P3,R,A2

- 8.1. Koliko puta koji od procesora pristupa memoriji? Za svaki pristup navesti razlog. [3 poena]
- 8.2. Koliki je Hit Rate za svaki od procesora (brojati i čitanje i upis, prikazati zbirno)? [3 poena]
- 8.3. Napisati stanja koherencije u svim procesorima i stanje memorije nakon svakog koraka (dovoljno je stanja ispisivati samo posle promene). [9 poena]
9. Sastaviti MPI program na jeziku C ili C++ koji vrši zamenu kolona matrice, tako da prva postane poslednja, druga preposlednja i tako redom. Proces sa rangom 0 (gospodar) u MPI svetu treba da obavlja svu komunikaciju sa korisnikom (unos matrice i ispis rezultata) i upravlja drugim procesima. Ostali procesi treba da dobiju po jednu vrstu matrice, izvrše obradu, vrate rezultat gospodaru, a zatim ponavljaju opisani postupak sve dok ih gospodar ne obavesti da nema više posla. Proces koji prvi pošalje rezultat prvi treba da dobije novu vrstu za obradu. Za prenos jedne vrste matrice koristiti izvedene tipove podataka.[15 poena]

Napomena:

U zadacima pretpostaviti da funkcije koje obavljaju potrebne ulazne i izlazne radnje već postoje, tako da za njih samo treba navesti prototipove i pozvati ih na odgovarajućim mestima u programskom kodu. Pretpostaviti da korisnik unosi sintaksno ispravne podatke.

Ukoliko u bilo kom pitanju ili zadatku nešto nije dovoljno precizno definisano, student treba da uvede razumnu pretpostavku, da je u okviru (da bi bila lakše prepoznata prilikom ocenjivanja) i da nastavi da izgrađuje preostali deo svog odgovora na temeljima uvedene pretpostavke.